

RESEARCH

Open Access

Generalized generating function with tucker decomposition and alternating least squares for underdetermined blind identification

Fanglin Gu^{1*}, Hang Zhang¹, Wenwu Wang² and Desheng Zhu¹**Abstract**

Generating function (GF) has been used in blind identification for real-valued signals. In this paper, the definition of GF is first generalized for complex-valued random variables in order to exploit the statistical information carried on complex signals in a more effective way. Then an algebraic structure is proposed to identify the mixing matrix from underdetermined mixtures using the generalized generating function (GGF). Two methods, namely GGF-ALS and GGF-TALS, are developed for this purpose. In the GGF-ALS method, the mixing matrix is estimated by the decomposition of the tensor constructed from the Hessian matrices of the GGF of the observations, using an alternating least squares (ALS) algorithm. The GGF-TALS method is an improved version of the GGF-ALS algorithm based on Tucker decomposition. More specifically, the original tensor, as formed in GGF-ALS, is first converted to a lower-rank core tensor using the Tucker decomposition, where the factors are obtained by the left singular-value decomposition of the original tensor's mode-3 matrix. Then the mixing matrix is estimated by decomposing the core tensor with the ALS algorithm. Simulation results show that (a) the proposed GGF-ALS and GGF-TALS approaches have almost the same performance in terms of the relative errors, whereas the GGF-TALS has much lower computational complexity, and (b) the proposed GGF algorithms have superior performance to the latest GF-based baseline approaches.

Keywords: Blind identification; Generalized generating function; Tensor decomposition; Tucker decomposition; Underdetermined mixtures

1. Introduction

Blind identification (BI) of linear mixtures has recently attracted intensive research interest in many fields of signal processing including blind source separation (BSS). This work is devoted to BI of underdetermined mixtures with complex sources. Underdetermined mixtures are commonly encountered in many practical applications, such as in the radio communication context, where the reception of more sources than sensors becomes increasingly possible with the growth in reception bandwidth. In these applications, one often has to also deal with complex sources. One reason is that the communication signals are usually complex-valued such as the quadrature-amplitude modulation (QAM) signal,

and minimum-shift keying (MSK) signal. Another reason is that frequency domain methods are often used for blind separation or identification from convolutive mixtures due to its computational efficiency [1,2], while the objective functions used in the frequency domain are usually defined on complex-valued variables.

A large number of methods for BI of underdetermined mixtures start from the assumption that the sources are sparse by nature (i.e., in its own domain such as the time domain) or could be made sparse in another domain (e.g., a transform domain). A predefined transform such as short-time Fourier transform (STFT) or a learned transform using, e.g., simultaneous codeword optimization (SimCO), is usually applied to sparsify the data [3,4] if the signal by nature is not sparse. Due to the sparsity of the sources, the scatter plot typically shows high signal values in the directions of the mixing vectors, which can be localized by using some clustering techniques [5,6]. It should

* Correspondence: gu.fanglin@gmail.com¹College of Communication Engineering, PLA University of Science & Technology, Nanjing 210007, People's Republic of China

Full list of author information is available at the end of the article

be noted that although some signals such as speech signals have some degree of sparsity in one domain or another, many other signals such as the majority of communication signals do not possess such a property. Hence, it is necessary to develop BI methods for the underdetermined mixtures that do not impose any sparsity constraint on the sources.

To this aim, many methods for BI of underdetermined mixtures turn to the use of various decomposition methods based on different data structures such as correlation [7,8] and higher-order cumulant [9-14] matrices. The main idea of these algorithms is to construct a tensor based on the cumulants of the observations and then to estimate the mixing matrix by the decomposition of such a tensor. This is notably the case for second-order blind identification of underdetermined mixtures (SOBIUM) [7], fourth-order blind identification of underdetermined mixtures (FOBIUM) [9], fourth-order-only blind identification (FOOBI) [10], FOOBI-2 [10], and blind identification of mixtures of sources using Redundancies in the daTa Hexacovariance matrix (BIRTH) [11,12] algorithms, which use second-order statistics tensors and fourth- and sixth-order cumulant tensors, respectively. A family of the methods named blind identification of over-complete mixtures of sources (BIOME) is proposed in [13], based on the even-order cumulants of the observations. However, all the methods proposed in [7-14] exploit only the statistical information contained in the data measured by second-order or higher-order statistics.

In order to exploit statistical information more effectively, a family of BI approaches was proposed in [15-18] by exploiting the statistical information with the characteristic function (CAF) or generating function (GF). In these works, the authors showed that the mixing matrix can be estimated up to trivial scaling and permutation indeterminacies by decomposing the tensor composed of partial derivatives of the GF. It is worth mentioning that the algorithms in [15-17] have been only applied to BI problems involving real-valued sources. In [18], the CAF approach was extended to the case of mixtures of complex-valued sources, which often occurs in digital communications. However, extra effort is required to obtain the correct real and imaginary combination of the mixing matrix since the real and imaginary parts of the mixing matrix are treated separately, leading to an increased computational cost due to the increased dimension of the matrix that needs to be processed. In this paper, we propose the Generalized Generating Function (GGF) to exploit the statistical information carried on the complex random variable. We show that the proposed GGF can exploit the statistical information carried on complex random variables in a more effective way than the GF presented in [18] due to the algebraic structure adopted

by GGF (as detailed in Algebraic structure based on generalized generating function). Furthermore, a simple method for the mixing matrix estimation is derived based on tensor decomposition where the tensor is composed of the Hessian matrices of the GGF of the observations.

The remainder of this paper is organized as follows. In Problem formulation the BI problem is formulated and relevant assumptions are presented. In Algebraic structure based on generalized generating function we firstly generalize the definition of the GF for complex-valued random variables and then derive the corresponding core equation for BI. In Blind identification based on tensor decomposition, the GGF-ALS and GGF-TALS approaches are developed for the estimation of the mixing matrix. In the GGF-ALS algorithm, the mixing matrix is estimated by directly decomposing the tensor, constructed from the Hessian matrices of the second GGF of the observations, using the alternating least squares (ALS) algorithm. In the GGF-TALS algorithm, the Tucker decomposition is firstly applied to convert the original tensor to a lower-order core tensor, then the mixing matrix is obtained by decomposing the core tensor with the ALS algorithm. Furthermore, the factors of Tucker decomposition are obtained by the left singular vectors of the original tensor's mode-3 matrix. Computer simulations are used to illustrate the performance of the proposed GGF approaches in Simulations and analysis. Finally, the paper is concluded in Conclusions.

2. Problem formulation

Considering the following linear mixture model

$$\mathbf{z}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{w}(t) \quad (1)$$

where the stochastic vector $\mathbf{z}(t) \in \mathbb{C}^Q$ represents the observation signals, $\mathbf{s}(t) \in \mathbb{C}^P$ contains the unobserved source signals, and $\mathbf{w}(t) \in \mathbb{C}^Q$ denotes additive noise. From now on, the noise $\mathbf{w}(t)$ is simply ignored for convenience, except when running computer experiments. The unknown mixing matrix $\mathbf{A} \in \mathbb{C}^{Q \times P}$ characterizes the way that the sources are acquired by the sensors. BI aims to estimate the mixing matrix from the observations based on the assumption that the source signals are statistically independent. The mixing matrix obtained may in turn be used to estimate the original source signals from the observations. In addition, we make the following assumptions:

- (i). The mixing matrix \mathbf{A} is of full (row) rank.
- (ii). The number P of sources is known.
- (iii). The number of sensors is smaller than the number of sources, i.e., $Q < P$.

3. Algebraic structure based on generalized generating function

3.1. Core equation based on generalized generating function

For a real stochastic vector $\mathbf{x} \in \mathbb{R}^Q$, the GF $\phi_x(\mathbf{u})$ obtained by dropping the term of the square root of (-1) in the exponent of a CAF is defined as

$$\phi_x(\mathbf{u}) = E[\exp(\mathbf{u}^T \mathbf{x})], \mathbf{u} \in \mathbb{R}^Q, \quad (2)$$

where $\mathbf{u} \in \mathbb{R}^Q$ is an arbitrary vector referred to as a processing point [15], and $E[\cdot]$ denotes an expectation operator. Nevertheless, both the observation vector \mathbf{z} and the mixing matrix \mathbf{A} discussed in this paper belong to the complex field. Hence, a definition of GF for complex variables is required. One such definition has been presented in [18] as

$$\begin{aligned} \phi_z(\Re(\mathbf{u}), \Im(\mathbf{u})) &= E[\exp(\Re(\mathbf{u}^H \mathbf{z}))] \\ &= E[\exp(\Re(\mathbf{u}^T(\mathbf{u}\Re(\mathbf{z}) + \Im^T(\mathbf{u})\Im(\mathbf{z})))], \mathbf{u} \in \mathbb{C}^Q, \end{aligned} \quad (3)$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote taking the real and imaginary parts from their arguments (i.e., complex-valued vectors) to form a real-valued vector of the same dimension. It is actually defined by assimilating \mathbb{C} to \mathbb{R}^2 . Thus the GF of a complex variable in (3) is defined as a function of the real and imaginary parts. In this paper, we generalize the definition of GF for real stochastic vector in (2) to the following complex form

$$\psi_z(\mathbf{u}) = E[\exp(\mathbf{u}^H \mathbf{z})], \mathbf{u} \in \mathbb{C}^Q \quad (4)$$

Note that the statistical information exploited by GF/GGF is related to the number of processing points. Theoretically, a complete statistical description of the probability density function requires the evaluation of the GF/GGF at all (infinitely many) possible processing points. However, this often becomes computationally infeasible. In practice, such statistical information is obtained approximately by the evaluation of GF/GGF at a finite number of processing points. Hence, in comparison with the GF presented in [18], the GGF defined in (4) can exploit the statistical information carried on the complex variables more effectively when the number of the processing points stays the same, thanks to the incorporation of the imaginary part of the exponent to the function. Furthermore, as compared with the use of the GF in (3), using the GGF in (4) offers a simpler way for the estimation of the mixing matrix due to the exploitation of an elegant algebraic structure.

Now, replacing \mathbf{z} by its model and neglecting the noise contribution yield

$$\psi_z(\mathbf{u}) = E[\exp(\mathbf{u}^H \mathbf{A} \mathbf{s})] = \psi_s(\mathbf{A}^H \mathbf{u}), \mathbf{u} \in \mathbb{C}^Q. \quad (5)$$

Defining $\phi_z(\mathbf{u}) = \log \psi_z(\mathbf{u})$, which is often referred to as the 'second' GGF, and using the source independence

property, the second GGF of the observations can be rewritten as

$$\phi_z(\mathbf{u}) = \sum_{p=1}^P \phi_{s_p}(\mathbf{a}_p^H \mathbf{u}) \quad (6)$$

Consequently, by calculating the derivative of the conjugate gradient of $\phi_z(\mathbf{u})$ with respect to \mathbf{u} (more details can be found in Appendix 1), we can obtain the following core equation for the Hessian matrix $\Psi_z(\mathbf{u})$,

$$\Psi_z(\mathbf{u}) = \mathbf{A} \Psi_s(\mathbf{A}^H \mathbf{u}) \mathbf{A}^H \quad (7)$$

with

$$\Psi_z(\mathbf{u}) = \nabla_{\mathbf{u}^T} [\nabla_{\mathbf{u}^*} \phi_z(\mathbf{u})] = \frac{\partial}{\partial \mathbf{u}^T} \left[\frac{\partial \phi_z(\mathbf{u})}{\partial \mathbf{u}^*} \right], \quad (8)$$

where $(\cdot)^*$ denotes the conjugate operator. It is necessary to point out that $\Psi_s(\mathbf{A}^H \mathbf{u})$ is a diagonal matrix (more details can be found in Appendix 2).

3.2. Estimating $\Psi_z(\mathbf{u})$

In this subsection, we discuss how to consistently estimate the Hessian matrix $\Psi_z(\mathbf{u})$. Under the ergodicity assumption, the mean value of a random variable can be estimated by a time average. Hence, we can estimate the GGF of the observation vector as

$$\hat{\psi}_z(\mathbf{u}) = \frac{1}{T} \sum_{t=1}^T \exp(\mathbf{u}^H \mathbf{z}(t))$$

The conjugate gradient $\hat{\zeta}_z(\mathbf{u}) = \partial \psi_z(\mathbf{u}) / \partial \mathbf{u}^*$ which is a $Q \times 1$ vector can be estimated by

$$\hat{\zeta}_z(\mathbf{u}) = \frac{1}{T} \sum_{t=1}^T \exp(\mathbf{u}^H \mathbf{z}(t)) \mathbf{z}(t)$$

Similarly, the gradient $\hat{\zeta}_z(\mathbf{u}) = \partial \zeta_z(\mathbf{u}) / \partial \mathbf{u}^T$ which is a $Q \times Q$ matrix can be estimated by

$$\hat{\zeta}_z(\mathbf{u}) = \partial \frac{\hat{\zeta}_z(\mathbf{u})}{\partial \mathbf{u}^T} = \frac{1}{T} \sum_{t=1}^T \exp(\mathbf{u}^H \mathbf{z}(t)) \mathbf{z}(t) \mathbf{z}^H(t)$$

Based on the above analysis, the Hessian matrix $\Psi_z(\mathbf{u})$ of the second GGF $\phi_z(\mathbf{u})$ can be obtained as

$$\hat{\Psi}_z(\mathbf{u}) = \frac{\hat{\zeta}_z(\mathbf{u})}{\hat{\psi}_z(\mathbf{u})} - \frac{\hat{\zeta}_z(\mathbf{u}) \hat{\zeta}_z^H(\mathbf{u})}{\hat{\psi}_z^2(\mathbf{u})}$$

4. Blind identification based on tensor decomposition

In this section, the GGF-ALS and GGF-TALS algorithms are developed for the estimation of the mixing matrix. In the GGF-ALS algorithm, the mixing matrix is estimated by decomposing the tensor which is formed of the Hessian matrices of the GGF of the observations. An improved version of the GGF-ALS algorithm, i.e., the GGF-TALS algorithm, is also developed, where the

Tucker decomposition is firstly employed to convert the original tensor as used in the GGF-ALS algorithm to a lower-rank core tensor, and the mixing matrix is then estimated by decomposing the core tensor with the ALS algorithm.

4.1. The GGF-ALS algorithm

Evaluating the Hessian matrices $\Psi_z(\mathbf{u})$ of GGF at a series of processing points $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K$ and using Equation (7), one can obtain the following joint diagonalization (JD) problem

$$\begin{cases} \Psi_z(\mathbf{u}_1) = \mathbf{A}\Psi_s(\mathbf{A}^H\mathbf{u}_1)\mathbf{A}^H \\ \vdots \\ \Psi_z(\mathbf{u}_K) = \mathbf{A}\Psi_s(\mathbf{A}^H\mathbf{u}_K)\mathbf{A}^H \end{cases} \quad (9)$$

in which $\Psi_s(\mathbf{A}^H\mathbf{u}_k)$ is diagonal, $k = 1, \dots, K$. The problem we need to address is to estimate the mixing matrix \mathbf{A} based on the set $\{\Psi_z(\mathbf{u}_1), \dots, \Psi_z(\mathbf{u}_K)\}$. For the determined/overdetermined case, it is obvious that JD methods, such as the AC-DC method [19], can be used to estimate the mixing matrix. However, this method does not work when $Q < P$ i.e., in the underdetermined case.

As shown in [7], the JD problem (9) can be seen as a particular case of the parallel factor (PARAFAC) decomposition, also known as canonical decomposition (CAND), of the third-order tensor $\mathcal{M} \in \mathbb{C}^{Q \times Q \times K}$ built by stacking the K matrices $\Psi_x(\mathbf{u}_k)$ along the third mode. Specifically, the tensor $\mathcal{M} \in \mathbb{C}^{Q \times Q \times K}$ is built by stacking $\Psi_z(\mathbf{u}_1), \Psi_z(\mathbf{u}_2), \dots, \Psi_z(\mathbf{u}_K)$ as follows: $(\mathcal{M})_{ijk} = (\Psi_z(\mathbf{u}_k))_{ij}$, $i = 1, \dots, Q$, $j = 1, \dots, Q$, $k = 1, \dots, K$. Define a matrix $\mathbf{D} \in \mathbb{C}^{K \times P}$ by $(\mathbf{D})_{kl} = (\Psi_s(\mathbf{A}^H\mathbf{u}_k))_{ll}$, $l = 1, \dots, P$, $k = 1, \dots, K$. Then we have

$$m_{ijk} = \sum_{l=1}^P a_{il} a_{jl}^* d_{kl}, \quad (10)$$

which we write as

$$\mathcal{M} = \sum_{l=1}^P \mathbf{a}_l \circ \mathbf{a}_l^* \circ \mathbf{d}_l, \quad (11)$$

where \circ denotes the tensor outer product, and \mathbf{a}_l and \mathbf{d}_l represent the l th column of \mathbf{A} and \mathbf{D} , respectively. In this way, the mixing matrix \mathbf{A} can be estimated by solving the following problem. Given the third-order tensor $\mathcal{M} \in \mathbb{C}^{Q \times Q \times K}$, we can compute its CAND with P components of the rank-one tensors that best approximates \mathcal{M} , i.e.,

$$\min_{\mathbf{A}, \mathbf{D}} \left\| \mathcal{M} - \sum_{p=1}^P \mathbf{a}_p \circ \mathbf{a}_p^* \circ \mathbf{d}_p \right\|_F^2, \quad (12)$$

where $\|\cdot\|_F$ is the Frobenius norm.

Several algorithms exist for the computation of tensor decomposition. The standard way for computing the tensor decomposition is by using an 'ALS' algorithm [20]. Several improved versions, such as the enhanced

line search (ELS) [21] and extrapolating search direction (ESD) [22], are proposed to accelerate the rate of convergence of the ALS. Hence, the ALS is chosen here to compute the CAND.

To a large extent, the practical importance of tensor decomposition stems from its uniqueness properties. It is clear that the tensor decomposition can only be unique up to a permutation of the rank-1 terms and scaling of the factors of the rank-1 terms. Therefore, we consider the tensor decomposition (11) as essentially unique if any other matrix pair \mathbf{A}' and \mathbf{D}' that satisfies (11) is related to \mathbf{A} and \mathbf{D} via

$$\mathbf{A} = \mathbf{A}'\mathbf{P}\mathbf{\Delta}_1, \quad \mathbf{D} = \mathbf{D}'\mathbf{P}\mathbf{\Delta}_2 \quad (13)$$

with $\mathbf{\Delta}_1, \mathbf{\Delta}_2 \in \mathbb{C}^{P \times P}$ being diagonal matrices, satisfying $\mathbf{\Delta}_1\mathbf{\Delta}_1\mathbf{\Delta}_2 = \mathbf{I}$, and $\mathbf{P} \in \mathbb{R}^{P \times P}$ being a permutation matrix.

The k -rank. The Kruskal rank or k -rank of a matrix \mathbf{A} , denoted by κ_A , is the maximal number λ such that any set of λ columns of \mathbf{A} is linearly independent.

Theorem 1. *The tensor decomposition of (11) is essentially unique if [23]*

$$2\kappa_A + \kappa_D \geq 2(P + 1) \quad (14)$$

We call a property generic when it holds with probability one. Generically, the mixing matrix is of full rank and of full k -rank when the parameters it involves are drawn from continuous probability densities. Hence, in practice, $\kappa_A = \min(Q, P)$ and $\kappa_D = \min(K, P)$.

In summary, we come to the following conclusion: when $Q \geq P$, $P \geq 2$, then the generic essential uniqueness is guaranteed for $K \geq 2$; when $Q < P$ and if $K \geq P$, then the generic essential uniqueness is guaranteed for $P \leq 2Q - 2$, if $K < P$, then the generic essential uniqueness is guaranteed for $P < Q - 1 + K/2$.

4.2. The GGF-TALS algorithm

In order to ensure the GGF-ALS algorithm has robust performance, a large value is often chosen for K in the tensor $\mathcal{M} \in \mathbb{C}^{Q \times Q \times K}$. Nevertheless, this will lead to a heavy computational load. To reduce the computational complexity, the Tucker decomposition [23-25] is firstly applied to represent the tensor \mathcal{M} as a lower-rank core tensor, whose size is much smaller than the tensor \mathcal{M} . Then the ALS algorithm is used to perform the CAND of the core tensor. In this way, the computational complexity can be reduced dramatically.

Matricization. Matricization, also known as unfolding or flattening, is the process of turning an N -way tensor into a matrix. The mode- n matricization of a tensor $\mathcal{F} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted by $\mathbf{F}_{(n)}$ which arranges the mode- n fibers to be the columns of the resulting

matrix, that is, the tensor element (i_1, \dots, i_N) is mapped to the matrix element (i_n, j) where

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^N (i_k - 1) J_k$$

with

$$J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m$$

The n -rank. Let \mathcal{F} be an N th-order tensor of size $I_1 \times I_2 \times \dots \times I_N$. Then the n -rank of \mathcal{F} , denoted by $\text{rank}_n(\mathcal{F})$, is the column rank of $\mathbf{F}_{(n)}$. In other words, if we let $r_n = \text{rank}_n(\mathcal{F})$ for $n = 1, \dots, N$, then we can say that \mathcal{F} is a rank r_1, \dots, r_N tensor.

The n -mode product. The n -mode (matrix) product of a tensor $\mathcal{F} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$ with a matrix $\mathbf{U}^{J \times I_n}$ is denoted by $\mathcal{F} \times_n \mathbf{U}$ and is of size $I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N$. Elementwise, we have

$$(\mathcal{F} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} f_{i_1 \dots i_n \dots i_N} u_{j i_n}$$

Tucker decomposition is a form of higher-order principal component analysis (PCA). It decomposes a tensor into a core tensor multiplied by a matrix along each mode as shown in Figure 1. Thus, in the three-way case where $\mathcal{F} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$, we have

$$\begin{aligned} \mathcal{F} &= \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} + \mathcal{U} \\ &= \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} g_{j_1 j_2 j_3} \left(\mathbf{a}_{j_1}^{(1)} \circ \mathbf{a}_{j_2}^{(2)} \circ \mathbf{a}_{j_3}^{(3)} \right) + \mathcal{U}, \end{aligned} \quad (15)$$

where $\mathcal{G} \in \mathbb{C}^{J_1 \times J_2 \times J_3}$ is the core tensor; $\mathbf{A}^{(1)} \in \mathbb{C}^{I_1 \times J_1}$, $\mathbf{A}^{(2)} \in \mathbb{C}^{I_2 \times J_2}$, $\mathbf{A}^{(3)} \in \mathbb{C}^{I_3 \times J_3}$ are the factor matrices (which are usually column unitary for real-valued data) and can

be thought of as the principal components in each mode; and $\mathcal{U} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ represents errors or noise.

With the help of Tucker decomposition, the tensor \mathcal{M} , which is composed of the Hessian matrices $\Psi_z(\mathbf{u}_k)$ of the second GGF of the observations, can be compressed. Since the mixing matrix \mathbf{A} is of full rank and the Hessian matrices $\Psi_s(\mathbf{A}^H \mathbf{u}_k)$ of the second GGF of the sources are diagonal, the mode- n ($n = 1, 2, 3$) matrices of tensor \mathcal{M} are $\mathbf{M}_{(1)} \in \mathbb{C}^{Q \times QK}$, $\mathbf{M}_{(2)} \in \mathbb{C}^{Q \times QK}$, and $\mathbf{M}_{(3)} \in \mathbb{C}^{K \times QQ}$ respectively. Meanwhile, the n -rank of the $\mathbf{M}_{(1)}$ and $\mathbf{M}_{(2)}$ is

$$\begin{aligned} \text{rank}_1(\mathcal{M}) &= \text{rank}_2(\mathcal{M}) = \text{rank}(\mathbf{M}_{(1)}) \\ &= \text{rank}(\mathbf{M}_{(2)}) = Q \end{aligned} \quad (16)$$

On the other hand, if we assume $\text{rank}_3(\mathcal{M}) = L$ and $L \leq K$, then \mathcal{M} is a rank- (Q, Q, L) tensor. Thus, the Tucker decomposition of tensor \mathcal{M} is the so-called Tucker1 decomposition [23]

$$\mathcal{M} = \mathcal{T} \times_1 \mathbf{I} \times_2 \mathbf{I} \times_3 \mathbf{G} \quad (17)$$

where $\mathcal{T} \in \mathbb{C}^{Q \times Q \times L}$ is the core tensor, $\mathbf{I} \in \mathbb{R}^{Q \times Q}$ is an identity matrix, and $\mathbf{G} \in \mathbb{C}^{K \times L}$ is a column-unitary matrix. This is equivalent to a standard two-dimensional PCA since

$$\mathbf{M}_{(3)} = \mathbf{G} \times \mathbf{T}_{(3)}, \quad (18)$$

where $\mathbf{T}_{(3)} \in \mathbb{C}^{L \times QQ}$ is the mode-3 matrix of core tensor \mathcal{T} . It is obvious that (18) corresponds to the PCA of $\mathbf{M}_{(3)}$. Therefore, $\mathbf{G} \in \mathbb{C}^{K \times L}$ consists of the L -leading left singular vectors of $\mathbf{M}_{(3)}$. Since $K > L$ and \mathbf{G} is a column-unitary matrix, it is straightforward to derive

$$\mathbf{T}_{(3)} = \mathbf{G}^H \times \mathbf{M}_{(3)} \quad (19)$$

Therefore, the core tensor can be obtained by

$$\mathcal{T} = \mathcal{M} \times_1 \mathbf{I} \times_2 \mathbf{I} \times_3 \mathbf{G}^H \quad (20)$$

Because the first and second factors of the Tucker decomposition in (17) are identity matrices, the core tensor $\mathcal{T} \in \mathbb{C}^{Q \times Q \times L}$ is also a symmetric tensor [23], as

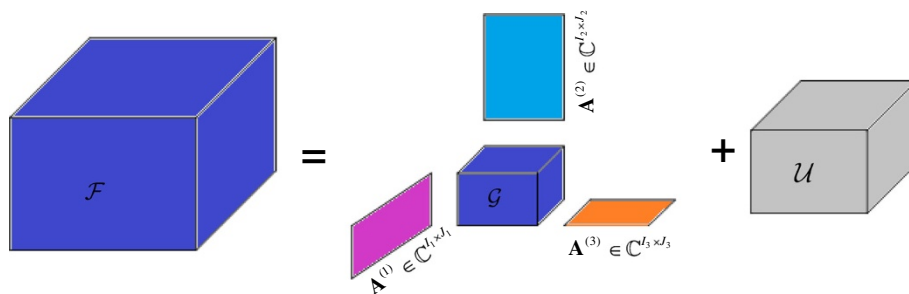


Figure 1 The Tucker decomposition of a three-way tensor \mathcal{F} . For data compression, the size of the core tensor \mathcal{G} is smaller than that of \mathcal{F} ($J_1 \leq I_1$, $J_2 \leq I_2$, and $J_3 \leq I_3$), and $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$, and $\mathbf{A}^{(3)}$ are thin matrices.

for the tensor $\mathcal{M} \in \mathbb{C}^{Q \times Q \times K}$ and the CAND of the core tensor is as follows

$$\mathcal{T} = \sum_{l=1}^P \mathbf{a}_l \circ \mathbf{a}_l^* \circ \mathbf{d}_l, \quad (21)$$

where \mathbf{a}_l and \mathbf{d}_l are the l th column of \mathbf{A} and \mathbf{D} , respectively. The CAND process can also be implemented by the ALS algorithm. Nevertheless, our goal is to estimate the mixing matrix \mathbf{A} , whereas the CAND of core tensor $\mathcal{T} \in \mathbb{C}^{Q \times Q \times L}$ only conduces \mathbf{A} . Hence, it is necessary to derive the mixing matrix \mathbf{A} based on \mathbf{A} . Since the first and second factors of Tucker decomposition (17) are identity matrices, it is straightforward to derive

$$\mathbf{A} = \mathbf{I}\mathbf{A} = \mathbf{A} \quad (22)$$

4.3. Computational analysis

In this subsection, we aim at giving an insight into the numerical complexity of the proposed algorithm. For the GGF-ALS algorithm, the ALS algorithm is directly used to decompose the tensor $\mathcal{M} \in \mathbb{C}^{Q \times Q \times K}$; therefore, the computational complexity is $O(3PKQ^2 + QKP^2 + Q^2P^2)$ per iteration. For the GGF-TALS algorithm, the computational complexity is dominated by the Tucker decomposition of the original tensor $\mathcal{M} \in \mathbb{C}^{Q \times Q \times K}$ (realized by the singular-value decomposition (SVD) of the mode-3 matrix $\mathbf{M}_{(3)}$) and the decomposition of the core tensor $\mathcal{T} \in \mathbb{C}^{Q \times Q \times L}$ using the ALS algorithm. Therefore, the computational complexities for these two operations per iteration are $O(Q^6)$ and $O(3PLQ^2 + QLP^2 + Q^2P^2)$, respectively. Table 1 shows the computational complexity for the GGF-ALS, and GGF-TALS methods.

It is worth mentioning that a large value is usually chosen for the number of processing points K in order to accumulate sufficient statistical information from data, while the rank of the core tensor order L is often chosen to be much smaller than K . For example, $K = 100$, $L = 8$ are chosen typically in our simulations as shown in the next section. The SVD is required to be calculated only once, whereas the ALS algorithm needs multiple iterations to achieve convergence, for example, 1,000 iterations as in our experiments. Moreover, the number of sources P and the number of sensors Q are usually small, for example, $P = 4$, $Q = 3$ in our simulations. For these reasons, it can be readily derived

from Table 1 that the computational cost of the GGF-TALS algorithm is, in practice, much lower than that of the GGF-ALS algorithm (note that the complexity of SVD becomes negligible in this case).

5. Simulations and analysis

In this section, simulations are provided to illustrate the performance of the proposed GGF approaches for underdetermined mixtures of complex sources. The performance of the tested algorithms is evaluated and compared in terms of the relative error performance index (PI) versus the sample size and the signal-to-noise ratio (SNR) of the observations. Here the relative error PI is defined as [7] $PI = E\{\|\mathbf{A} - \hat{\mathbf{A}}\| / \|\mathbf{A}\|\}$, in which the norm is the Frobenius norm and $\hat{\mathbf{A}}$ represents the optimally ordered and scaled estimate of the mixing matrix \mathbf{A} .

The experiments refer to the scenario that $P = 4$ narrowband source signals are received by a uniform circular array (UCA) with $Q = 3$ identical sensors of radius R_a . Considering a free space propagation model, the entries of the mixing matrix \mathbf{A} are given by

$$a_{qp} = \exp\left(2\pi j\left(\alpha_q \cos(\theta_p) \cos(\phi_p) + \beta_q \cos(\theta_p) \sin(\phi_p)\right)\right)$$

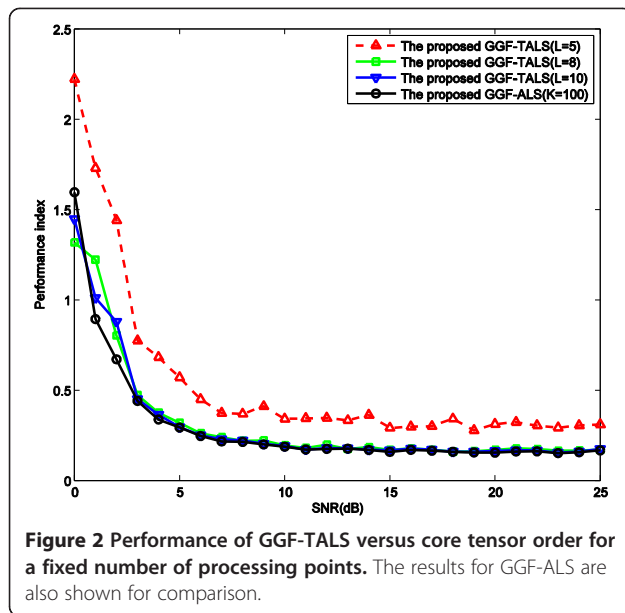
where $\alpha_q = (R_a/\lambda)\cos(2\pi(q-1)/Q)$, $\beta_q = (R_a/\lambda)\sin(2\pi(q-1)/Q)$, and $j = \sqrt{-1}$. We have $R_a/\lambda = 0.55$. The direction of arrival (DOA) of the different sources are given by $\theta_1 = 3\pi/10$, $\theta_2 = 3\pi/10$, $\theta_3 = 2\pi/5$, $\theta_4 = 0$ and $\phi_1 = 7\pi/10$, $\phi_2 = 9\pi/10$, $\phi_3 = 3\pi/5$, $\phi_4 = 4\pi/5$. The sources are unit-variance 4-QAM with a uniform distribution, shaped by a raised cosine pulse shaping filter with a roll-off $\rho = 0.3$. All sources have the same symbol duration $T = 4T_e$, where T_e is the sample period. The observations are contaminated by additive zero-mean complex Gaussian noise.

First, we compare the performance of the GGF-ALS algorithm with that of the GGF-TALS algorithm and investigate the influence of the rank of the core tensor L on the performance of the GGF-TALS algorithm. To this end, the following two simulation experiments are conducted. In the first simulation, we evaluate the performance of the GGF-TALS algorithm with different core tensor rank L for a fixed number of processing points K , and compare it with GGF-ALS with the same number of processing points. In this simulation, K is chosen to be 100 for the GGF-ALS algorithm, both the real and imaginary parts of processing points are randomly drawn from $[-1; 1]$, the SNR of the observations ranges from 0 to 25 dB, and the number of samples is 4,000. The core tensor rank for the GGF-TALS algorithm is chosen as $L = 10, 8, 5$, respectively. The threshold value described in (12) to stop the ALS algorithm is 10^{-5} , and 100 Monte Carlo experiments are run.

The average performance of the GGF-TALS algorithm versus the core tensor rank for a fixed number of processing points is shown in Figure 2. We can see that the

Table 1 Computational complexity of GGF-ALS, GGF-TALS methods

Method	Operation	Complexity
GGF-ALS	ALS (one iteration)	$O(3PKQ^2 + QKP^2 + Q^2P^2)$
GGF-TALS	SVD	$O(Q^6)$
	ALS (one iteration)	$O(3PLQ^2 + QLP^2 + Q^2P^2)$



performance curves of the GGF-TALS and that of the GGF-ALS almost coincide when the rank of the core tensor is larger than 8, whereas the performance of the GGF-TALS algorithm slightly deteriorates when the rank of the core tensor is less than 8. This indicates that the GGF-TALS algorithm maintains the identification accuracy offered by GGF-ALS despite the fact that the core tensors used by GGF-TALS have a much lower rank than that of the original tensor used by GGF-ALS.

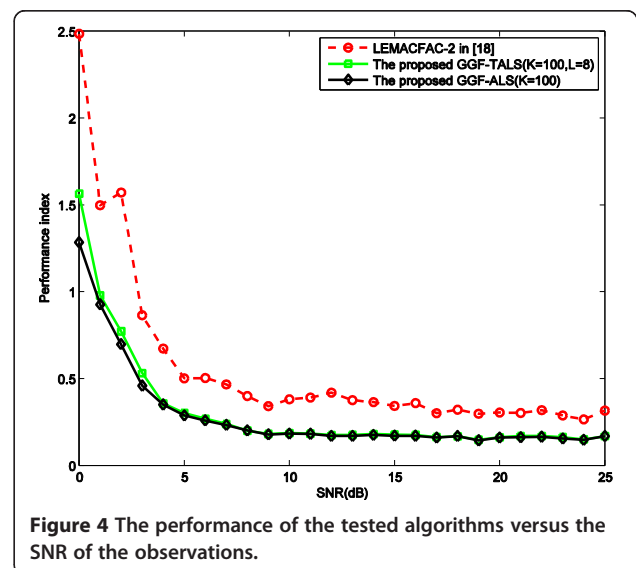
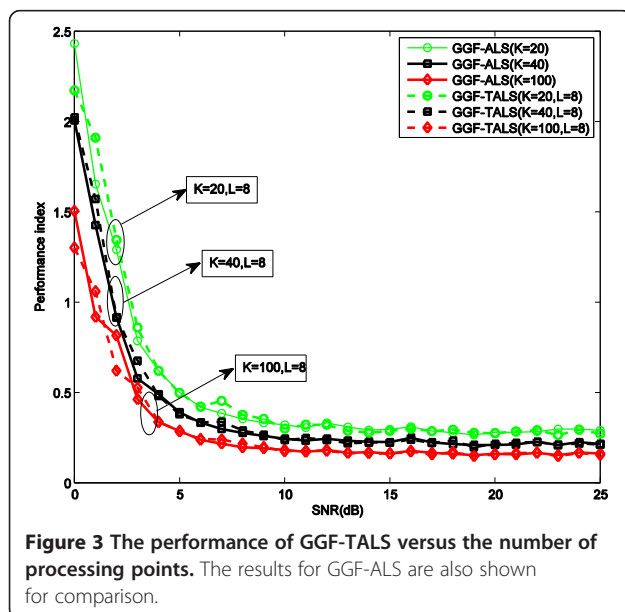
In the second experiment, we investigate the performance of the GGF-TALS algorithm with different

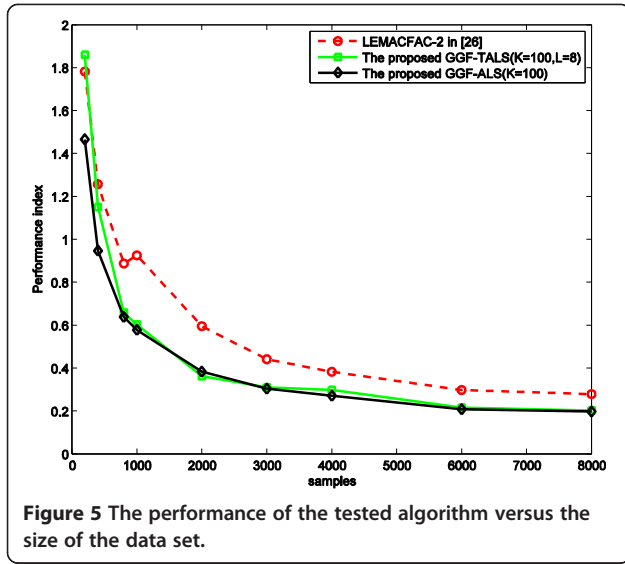
number of processing points K when the rank of the core tensor L is fixed. The simulation conditions are the same as those in the previous simulation except the number of processing points for the GGF-ALS algorithm which are 20, 40, and 100 respectively, and the rank of the core tensor for the GGF-TALS which is 8. The average performance of the GGF-TALS versus the number of processing points for the fixed number of the core tensor rank is shown in Figure 3.

We can see that the average performance of GGF-TALS is consistent with that of GGF-ALS when both algorithms use the same number of processing points. For instance, when the number of processing points is 100, the performance of GGF-TALS is close to the performance of GGF-ALS, so is for $K = 40$. Therefore, the GGF-TALS algorithm consistently offers similar performance to GGF-ALS when the number of processing points is varied.

Second, we compare the performance of the GGF-ALS and GGF-TALS algorithms with that of the latest GF-based BI algorithm. Here, the LEMACFAC-2 in [18] is chosen as the baseline algorithm. The number of processing points and the value of processing points for the GGF-ALS, GGF-TALS, and LEMACFAC-2 are the same. Specifically, the number of processing points is 100, both the real and imaginary parts of processing points are randomly drawn from $[-1; 1]$, and the rank of the core tensor is 8. The threshold value described in (12) to stop the LM algorithm [26] exploited in LEMACFAC-2 method is also 10^{-5} .

Figure 4 shows the PI of the tested algorithms as a function of the SNR when 4,000 samples are used. It can be seen from this figure that the performance of the GGF-ALS is consistent with that of the GGF-TALS, and both perform better than LEMACFAC-2. Figure 5 shows





the PI of the tested algorithms as a function of the number of data samples N when the SNR is equal to 20 dB. Again, the LEMACFAC-2 underperforms the GGF-ALS and the GGF-TALS algorithms, and the GGF-ALS and the GGF-TALS have almost the same performance. This confirms that the GGF algorithms perform better than the method based on the GF defined in (3) in exploiting the statistical information carried on the complex variables when using the same number of processing points.

6. Conclusions

We have presented two algorithms, namely, GGF-ALS and GGF-TALS, for blind identification from underdetermined mixtures of complex sources using the second GGF of the observations. In the GGF-ALS algorithm, the mixing matrix is estimated by directly using the ALS algorithm to decompose the tensor constructed from the Hessian matrices of the second GGF of the observations. The GGF-TALS algorithm is an improved version of the GGF-ALS algorithm, where the Tucker decomposition is first used to convert the original tensor into a lower-rank core tensor, and the mixing matrix is then estimated by applying the ALS algorithm to the core tensor. Simulation results have shown that (a) the proposed GGF-ALS and GGF-TALS approaches have almost the same performances in terms of the relative errors, whereas the GGF-TALS has a much lower computational complexity, and (b) the proposed GGF algorithms have superior performance to the latest GF-based BI approaches, since the GGF algorithms can exploit the statistical information carried on complex variables in a more effective way.

Appendix 1

In this appendix, we show the computational details of the core equation in Equation 7. First, the differentiation of (6) with respect to \mathbf{u}^* gives

$$\begin{aligned}\frac{\partial \varphi_z(\mathbf{u})}{\partial \mathbf{u}^*} &= \sum_{p=1}^P \frac{\partial \varphi_{s_p}(\mathbf{a}_p^H \mathbf{u})}{\partial \mathbf{u}^*} = \sum_{p=1}^P \frac{\partial \varphi_{s_p}(\mathbf{a}_p^H \mathbf{u})}{\partial (\mathbf{a}_p^H \mathbf{u})^*} \frac{\partial (\mathbf{a}_p^H \mathbf{u})^*}{\partial \mathbf{u}^*} \\ &= \sum_{p=1}^P \frac{\partial \varphi_{s_p}(\mathbf{a}_p^H \mathbf{u})}{\partial (\mathbf{a}_p^H \mathbf{u})^*} \frac{\partial (\mathbf{a}_p^H \mathbf{u})^H}{\partial \mathbf{u}^*} = \sum_{p=1}^P \frac{\partial \varphi_{s_p}(\mathbf{a}_p^H \mathbf{u})}{\partial (\mathbf{a}_p^H \mathbf{u})^*} \mathbf{a}_p\end{aligned}\quad (23)$$

Second, the differentiation of (23) with respect to \mathbf{u}^T gives

$$\begin{aligned}\frac{\partial \varphi_z(\mathbf{u})}{\partial \mathbf{u}^T \partial \mathbf{u}^*} &= \sum_{p=1}^P \frac{\partial \varphi_{s_p}(\mathbf{a}_p^H \mathbf{u})}{\partial (\mathbf{a}_p^H \mathbf{u})^T} \frac{\partial (\mathbf{a}_p^H \mathbf{u})^T}{\partial \mathbf{u}^T} \mathbf{a}_p \\ &= \sum_{p=1}^P \frac{\partial \varphi_{s_p}(\mathbf{a}_p^H \mathbf{u})}{\partial (\mathbf{a}_p^H \mathbf{u})^H} \frac{\partial (\mathbf{a}_p^H \mathbf{u})}{\partial \mathbf{u}^T} \mathbf{a}_p \\ &= \sum_{p=1}^P \frac{\partial \varphi_{s_p}(\mathbf{a}_p^H \mathbf{u})}{\partial (\mathbf{a}_p^H \mathbf{u})^H} \mathbf{a}_p \mathbf{a}_p^H\end{aligned}\quad (24)$$

In a more compact form, we can obtain the core equation

$$\Psi_z(\mathbf{u}) = \mathbf{A} \Psi_s(\mathbf{A}^H \mathbf{u}) \mathbf{A}^H \quad (25)$$

Appendix 2

Let $\phi_s(\tilde{\mathbf{u}})$ denote the characteristic function of the source signals $\mathbf{s}(t)$. Due to the statistical independence of the elements of $\mathbf{s}(t) = [s_1(t), \dots, s_P(t)]^T$, we have

$$\phi_s(\tilde{\mathbf{u}}) = \phi_{s_1}(\tilde{u}_1) \times \phi_{s_2}(\tilde{u}_2) \times \dots \times \phi_{s_P}(\tilde{u}_P) \quad (26)$$

where $\phi_{s_j}(\tilde{u}_j) = E[\exp(\tilde{u}_j^* s_j)]$, $j = 1, \dots, P$. Therefore, for the second characteristic function $\varphi_s(\tilde{\mathbf{u}})$, we have

$$\varphi_s(\tilde{\mathbf{u}}) = \varphi_{s_1}(\tilde{u}_1) + \varphi_{s_2}(\tilde{u}_2) + \dots + \varphi_{s_P}(\tilde{u}_P) \quad (27)$$

Consequently, the Hessian matrix $\Psi_s(\tilde{\mathbf{u}})$ can be easily obtained as

$$\begin{aligned}\Psi_s(\tilde{\mathbf{u}}) &= \nabla_{\tilde{\mathbf{u}}}^T \left[\nabla_{\tilde{\mathbf{u}}}^* \varphi_s(\tilde{\mathbf{u}}) \right] = \frac{\partial}{\partial \tilde{\mathbf{u}}^T} \left[\frac{\partial \varphi_s(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}^*} \right] \\ &= \text{diag} \left(\frac{\partial \phi_{s_1}(\tilde{u}_1)}{\partial \tilde{u}_1 \partial \tilde{u}_1^*}, \frac{\partial \phi_{s_2}(\tilde{u}_2)}{\partial \tilde{u}_2 \partial \tilde{u}_2^*}, \dots, \frac{\partial \phi_{s_P}(\tilde{u}_P)}{\partial \tilde{u}_P \partial \tilde{u}_P^*} \right)\end{aligned}\quad (28)$$

where $\text{diag}(\cdot)$ represents an operator forming a diagonal matrix by assigning its arguments to the entries of the main diagonal.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported in part by Natural Science Foundation of China under grant 61001106 and National Program on Key Basic Research Project of China under grant 2009CB320400, and Engineering and Physical Science Research Council of the UK under grant EP/H050000/1. The authors would like to thank the anonymous referees for their constructive comments for improving this paper and Dr. Mark Barnard for proofreading the manuscript.

Author details

¹College of Communication Engineering, PLA University of Science & Technology, Nanjing 210007, People's Republic of China. ²Department of Electronic Engineering, University of Surrey, Guildford GU2 7XH, UK.

Received: 2 November 2012 Accepted: 21 June 2013

Published: 1 July 2013

References

1. B Chen, AP Petropulu, Frequency domain blind MIMO system identification based on second- and higher order statistics. *IEEE Trans. Signal Process.* **49**(8), 1677–1688 (2001)
2. K Rahbar, JP Reilly, A frequency domain method for blind source separation of convolutive audio mixtures. *IEEE Trans. Speech Audio Process.* **13**(5), 832–844 (2005)
3. A Aissa-El-Bey, N Linh-Trung, K Abed-Meraim, A Belouchrani, Y Grenier, Underdetermined blind separation of nondisjoint sources in the time-frequency domain. *IEEE Trans. Signal Process.* **55**(3), 897–907 (2007)
4. W Dai, T Xu, W Wang, Simultaneous codeword optimization (SimCO) for dictionary update and learning. *IEEE Trans. Signal Process.* **60**(12), 6340–6353 (2012)
5. P Bofill, M Zibulevsky, Underdetermined blind source separation using sparse representations. *Signal Process.* **81**, 2353–2362 (2001)
6. C Févotte, SJ Godsill, A Bayesian approach for blind separation of sparse sources. *IEEE Trans. Audio Speech Lang. Process.* **14**(6), 2174–2188 (2006)
7. LD Lathauwer, J Castaing, Blind identification of underdetermined mixtures by simultaneous matrix diagonalization. *IEEE Trans. Signal Process.* **56**(3), 1096–1105 (2008)
8. P Tichavský, Z Koldovský, Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources. *IEEE Trans. Signal Process.* **59**(3), 1037–1047 (2011)
9. A Ferréol, L Albera, P Chevalier, Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM). *IEEE Trans. Signal Process.* **53**(5), 1640–1653 (2005)
10. LD Lathauwer, J Castaing, JF Cardoso, Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Trans. Signal Process.* **55**(6), 2965–2973 (2007)
11. L Albera, A Ferréol, P Comon, P Chevalier, *Sixth order blind identification of under-determined mixtures (BIRTH) of sources*. Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (Nara, 2003), pp. 909–914
12. ALF de Almeida, X Luciani, P Comon, *Blind identification of underdetermined mixtures based on the hexacovariance and higher-order cyclostationarity*. IEEE Workshop on Statistical Signal Processing (Cardiff, UK, 2009), pp. 669–672
13. L Albera, A Ferréol, P Comon, P Chevalier, Blind identification of over-complete mixtures of sources (BIOME). *Lin. Algebra Appl.* **391**, 1–30 (2004)
14. A Karfoul, L Albera, G Birot, Blind underdetermined mixture identification by joint canonical decomposition of HO cumulants. *IEEE Trans. Signal Process.* **58**(2), 638–649 (2010)
15. A Yeredor, Blind source separation via the second characteristic function. *Signal. Process.* **80**(5), 897–902 (2000)
16. E Eidinger, A Yeredor, Blind MIMO identification using the second characteristic function. *IEEE Trans. Signal Process.* **53**(11), 4067–4079 (2005)
17. P Comon, M Rajih, Blind identification of underdetermined mixtures based on the characteristic function. *Signal. Process.* **86**(9), 2271–2281 (2006)

18. X Luciani, ALF de Almeida, P Comon, Blind identification of underdetermined mixtures based on the characteristic function: the complex case. *IEEE Trans. Signal Process.* **59**(2), 540–553 (2011)
19. A Yeredor, Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Trans. Signal Process.* **50**(7), 1545–1553 (2002)
20. LD Lathauwer, BD Moor, J Vandewalle, On the best rank-1 and rank- (R_1, R_2, \dots, R_N) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.* **21**, 1324–1342 (2000)
21. D Nion, LD Lathauwer, An enhanced line search scheme for complex-valued tensor decompositions: application in DS-CDMA. *Signal Process.* **88**(3), 749–755 (2008)
22. Y Chen, D Han, L Qi, New ALS methods with extrapolating search directions and optimal step size for complex-valued tensor decompositions. *IEEE Trans. Signal Process.* **59**(12), 5888–5898 (2011)
23. TG Kolder, BW Bader, Tensor decompositions and applications. *SIAM Rev.* **51**(3), 455–500 (2009)
24. G Bergqvist, EG Larsson, The higher-order singular value decomposition: theory and an application. *IEEE Signal Process. Mag.* **27**(3), 151–154 (2010)
25. G Zhou, A Cichocki, Fast and unique Tucker decompositions via multiway blind source separation. *Bull. Pol. Acad. Sci.* **60**(3), 389–405 (2012)
26. P Comon, X Luciani, ALF de Almeida, Tensor decomposition, alternating least squares and other tales. *J. Chemometr.* **23**, 393–405 (2009)

doi:10.1186/1687-6180-2013-124

Cite this article as: Gu et al.: Generalized generating function with Tucker decomposition and alternating least squares for underdetermined blind identification. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:124.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com